

Generatori statičnih stranica

Table of Contents

1. Najpoznatiji generatori statičnih stranica	1
Zaključno	6
Izrada dokumentacije pomoću Jekyll-a i GitHub stranica	6
1. korak: instalacija Jekylla	6
2. korak: Lokalno izradite Jekyll blog	7
3. korak: Objava sadržaja na Github Pages	8
4. korak: Dodajte nove sadržaje na blog.	10
5. korak: Q&A sesija	10

Danas otvaramo temu statičnih generatora stranica (eng. **static site generators**). Temu dinamičnih generatora stranica ostavljamo za naredne susrete.

Statični generatori stranica dio su tzv. **Jamstack-a**, arhitekture za razvoj weba koja je usmjerena na izradu brzih, sigurnih i skalabilnih sjedišta. Arhitektura se zasniva na JavaScriptu koji se izvodi na klijentskom računalu, API-u aplikacijskom programskom sučelju i označenom sadržaju. Ključno načelo je da se svi označeni sadržaji prethodno renderiraju i na taj način brzo isporučuju korisnicima.

1. Najpoznatiji generatori statičnih stranica

1. Gatsby <https://www.gatsbyjs.com/>



Gatsby pripada u red najčešće korištenih generatora statičnih stranica. Krasi ga ogroman broj dostupnih *plugina*.

2. React Static <https://github.com/react-static/react-static>



React Static je generator izrađen na temelju React-a. Cilj mu je osigurati pravocrtan i kvalitetan razvoj web sjedišta zasnovan na jednostavnoj krivulji učenja.

3. Sapper <https://sapper.svelte.dev/>



Sapper nije samo generator stranica budući omogućava renderiranje sadržaja na serveru, definiranje krajnjih API točaka i izradu web aplikacija visokih performansi.

4. Nextjs <https://nextjs.org/>



Izrađen je na temelju React-a, trenutno najpopularnijeg JavaScript razvojnog okvira. Popularan je jer omogućuje izradu progresivnih web aplikacija i aplikacija koje se renderiraju na serveru.

5. Nuxtjs <https://nuxtjs.org/>



Nuxtjs podržava renderiranje na strani servera i generiranje stranica poput Nextjs. Za Nuxtjs vam je potreban Vue u pozadini Vue omogućuje automatski *routing* i to bi mogao biti razlog zbog kojeg ćete se odlučiti za Nuexjs. Usmjerenje podataka je problem s kojim se susrećemo u izradi tzv. jednostraničnih aplikacija (eng. *single page application*).

6. Jekyll <https://jekyllrb.com/>



Možda ste primijetili da su GitHub Pages podržane od strane Jekylla – sustava koji na jednostavn i brz način generira optimizirane stranice. Jekyll se koristi za izradu web sjedišta i blogova koji su nalik GitHub stranicama. Sustav koristi Markdown označiteljski jezik za oblikovanje teksta i jasnog

prikaza sjedišta. Za izradu predložaka koristi Liquid, a uz pomoć ugrađenih alata omogućava jednostavnu i brzu migraciju sadržaja s postojećih sjedišta.

7. **Hugo** <https://gohugo.io/>



Hugo donosi stotine atraktivnih tema koje možete odabrati za izradu vlastite statične stranice, a posebno ga krasi brzina izrade sjedišta. Hugo odabiru brojni korisnici koji žele izraditi sjedište bez složenih konfiguracijskih procedura.

8. **Vuepress** <https://vuepress.vuejs.org/>



VuePress

Vue-powered Static Site Generator

Vuepress je minimalistički generator statičnih stranica koji se koristi za izradu tehničke dokumentacije. Podržan je od strane Vue okvira uz pomoć kojeg se dolazi do atraktivnih predložaka, tema i pluginova.

9. **Docusaurus** <https://docusaurus.io/>

Docusaurus makes it easy to maintain **Open Source** documentation websites.



Ako ste usmjereni na pisanje sadržaja za blog ili izradu tehničke dokumentacije, Docusaurus preuzmi na sebe teret izrade i objave HTML datoteka. Docusaurus posjeduje zanimljivo rješenje za verzioniranje i time olakšava proces dokumentiranja.

10. Scully <https://scully.io/>

Static Angular sites, wicked fast.

Scully makes building, testing and deploying **Jamstack** apps extremely simple.

Scully možete koristiti za brzu izradu Angular aplikacija i statičnih web sjedišta. Scully prethodno renderira cijelo sjedište u HTML-u i CSS-u koje potom možete koristiti za izradu jednostraničnih aplikacija temeljenih na Angularu.

11. Eleventy <https://www.11ty.dev/>

The logo for Eleventy, featuring the text '11ty' in a large, white, sans-serif font on a dark background.

Eleventy "razgovara" s brojnih jezicima za izradu predložaka poput `.hbs`, `.md`, `.html`, `.ejs` i sl. Čak vam omogućava korištenje svih navedenih jezika u jednom projektu.

12. Docsify <https://docsify.js.org/#/>



docsify_{4.11.6}

A magical documentation site generator.

Docsify je sjajan alat za izradu dokumentacijskih web sjedišta. Uspješno parsira Markdown datoteke i prikazuje ih u HTML-u. Docsify podržava Vue komponente, što znači da iz možete ubaciti u Markdown datoteke i kasnije uspješno renderirati. Docsify nudi veliki broj gotovih tema i atraktivnih pogodnosti poput brojnih korisnih pluginova, offline korištenja, funkcije pretraživanja, prikaz emojija i sl.

13. Gridsome <https://gridsome.org/>

A Jamstack framework for Vue.js

Gridsome makes it easy for developers to build static generated websites & apps that are **fast by default** 🚀

Ako React ima Gatsby onda Vuejs ima svog "konja za trku" koji se odaziva na ime Gridsome. Gridsome je novi projekt koji svojim funkcionalnostima privlači pozornost korisničke zajednice. Ako se kod jednostraničnih aplikacija suočavate s problemom SEO-a, Vuejs je vrlo vjerojatno rješenje vaših problema. Gridsome nudi ogroman ekosustav dostupnih tema i pluginova koji pojednostavljaju izradu statičnih stranica.

14. Sculpin <https://sculpin.io/>

Sculpin

Static Site Generator



Sculpin je napisan u PHP-u, jednom od najpopularnijih programskih jezika današnjice. Poput većine drugih generatora, Sculpin uzima ulazne Markdown datoteke i konvertira ih u statične

HTML stranice i time pozitivno utječu na brzinu odziva web sjedišta. Svakako predstavlja prvi izbor svima kojima je PHP poznat ili blizak.

Zaključno

Statični generatori web stranica/sjedišta predstavljaju svojevrsni povratak u ranu fazu razvoja weba kada su "webmasteri" (op.a. danas ih zovemo "developeri") izrađivali tone statičnih HTML datoteka. Gledajući evolucijski ovo je pomak prema naprijed a ne unatrag. Statični generatori rješavaju brojne probleme s kojima se razvijatelji i stvaratelji sadržaja susreću pri korištenju sustava za dinamično generiranje stranica/sjedišta. Statične stranice štede vrijeme i novce, a jamče isporuku web aplikacija visokih performansi.

Izrada dokumentacije pomoću Jekyll-a i GitHub stranica

U nastavku ćemo se uloviti u koštac s izradom statičnih stranica pomoću Jekylla, te objavom istih pomoću Github Pages.

Kao što smo prethodno naveli, Jekyll je generator statičnih stranica koji omogućava izradu jednostavnih web sjedišta i blogova. Github Pages će u ovom kontekstu poslužiti kao host na kojem ćemo udomiti stranice izrađene pomoću Jekylla. Riječ je o vrlo jednostavnoj kombinaciji dvaju alata koji nam omogućuju besplatnu izradu i hosting sjedišta/stranica. U izradi sjedišta koristim Mac pa vas molim da se poslužite dostupnim online izvorima (Google!) kako biste razriješili situaciju na Linux/Windows računalima. Ova pokazna vježba pretpostavlja da ste savladali osnove Git-a, te da su vam poznati temeljni koncepti tehnologija koje ćemo danas koristiti (npr. Markdown, AsciiDoc i sl. označiteljski jezici).

1. korak: instalacija Jekylla

1. Otvorite terminal na Macu ili Linuxu, odnosno Powershell / VSC na Windowsima.
2. Prije instalacije Jekylla potrebno je provjeriti da li je već instalirana (ispravna verzija) Ruby programskog jezika (potrebna verzija 2.1. ili viša).

Za provjeru instalacije unesite naredbu:

```
ruby -v
```

3. Sad provjerite da li je instalirana ispravna **gem** verzija. Unesite naredbu:

```
gem -v
```

4. Nakon prethodnih provjera možete nastaviti s instalacijom Jekylla. Za instalaciju unesite naredbu:

```
gem install jekyll bundler
```

Ovime biste trebali dohvatiti sve datoteke potrebne za uspješan dovršetak instalacija.

5. Nakon instalacije provjerite instaliranu verziju Jekylla unosom naredbe

```
jekyll -v
```

2. korak: Lokalno izradite Jekyll blog

Naziv bloga koji ćemo koristiti za vježbu je SITblog i u nastavku ću vam demonstrirati postupak izrade.

1. Za izradu Jekyll bloga potrebno je otvoriti terminal i ukucati sljedeću naredbu

```
jekyll new SITblog
```

Ovime ćete izraditi novi direktorij na računalu s istim nazivom kao i naziv bloga.

2. Za uređivanje i objavu sadržaja koristit ćemo Visual Studio Code. Ovaj program omogućava korištenje izravno korištenje prozora s terminalom čime ubrzavamo upravljanje sadržajima. Svakako instalirajte VSC ukoliko to već niste učinili, a nakon toga u njemu otvorite svoj SITblog direktorij. Sad biste trebali vidjeti sadržaj direktorija slično kao i na Slici 2.
3. Prije nastavka uređivanja sadržaja bloga provest ćemo lokalnu objavu bloga kako bismo provjerili prikaz.

Otvorite terminal unutar VSC-a i unesite sljedeću naredbu:

```
bundle exec jekyll serve
```

Stranica bi vam se trebala otvoriti na lokalnom hostu na adresi: <http://127.0.0.1:4000/>

Ovo znači da ćete odabirom poveznice lokalno otvoriti svoj blog.

Provjerimo različite direktorije koje je Jekyll izradio i naučimo kako ih možemo uređivati.

1. **_posts:** direktorij u kojem se nalaze sve objave. Svaka objava bi trebala biti u zasebnoj markdown datoteci.
2. **_site:** ove datoteke su odgovorne za kompajliranje bloga. Različite objave, slike i stranice nalaze se u različitim direktorijima, a pomoću **_sites** direktorija sve ih privodite funkciji i objavljujete kao web sjedište.
3. **_config.yml:** ova datoteka sadrži varijable (metapodatke) sjedišta poput naslova, vašeg korisničkog imena, url-ove i sl.

4. **about.md**: ovo je markdown datoteka koja sadrži podatke za objavu u `about` sekciji.
5. **gemfile**: **gemfile** u kojem se nalaze tzv. *dependencies*, datoteke o kojima Jekyll ovisi u svom radu (npr. ovdje je zabilježena zadana tema 'Minima'). Takvu vrstu informacija specificira gem datoteka.
6. **index.md**: sadrži informacije koje su vidljive na naslovnici vašeg bloga.

Brza prilagodba bloga i dodavanje relevantnih informacija o vama i vašem sadržaju postiže se uređivanjem sljedećih datoteka:

- Naslov, opis i ostali detalji nalaze se u `_config.yml` datoteci
- Sadržaj u **about.md** datoteci
- Sadržaj blog objava se nalazi u unutar **_posts** direktorija. Probajte copy-pate postojeće datoteke kako biste izradili drugu blog objavu.

Now type `jekyll serve` to see these changes in your local host
(`http://127.0.0.1:4000/`)

Važne napomene:

1. Svaki put kad napravite promjenu sadržaja možete provjeriti rezultat unosom naredbe `jekyll serve` i nakon toga provjerite stanje na svom lokalnom hostu
2. Kada provjeravate promjene nećete biti u mogućnosti koristiti terminal. Stoga je potrebno izaći iz pokrenutog procesa odabirom tipkovničke kratice **Ctrl+C**.

To je sva "mudrost" vezana za uređivanje sadržaja. Slobodno se poigrajte s Jekyllom i izradite sadržaj prema želji i vlastitom nahođenju.

3. korak: Objava sadržaja na Github Pages

Sad kada ste naučili pokrenuti, postaviti i urediti blog, došlo je vrijeme za njegovu objavu kako biste ga podijelili sa svim zainteresiranim osobama.

Za javnu objavu bloga koristit ćemo Github stranice koje su u potpunosti besplatne!

1. Pretpostavljam da do danas već svi imate otvorene studentske Github račune!!!
2. Otvorite novi javni Github repozitorij i dajte mu isti naziv kao i lokalnom direktoriju ("SITblog").
3. Nakon uspješno otvaranja repozitorija trebala bi se otvoriti stranica s postavkama povezivanja lokalnog i online repozitorija.

U ovom koraku trebate inicijalizirati lokalni git repozitorij (ovaj korak pretpostavlja da ste lokalno već instalirali git! Svakako provjerite status instalacije). Za inicijalizaciju git repozitorija otvorite VSC i u terminalu unesite sljedeću naredbu:

```
git init
```


Nakon toga trebamo sve sadržaje bloga prebaciti u **gh-pages** ogranak (eng. *branch*). To je potrebno jer Github traži stranice baš u navedenom ogranku! Da biste ste datoteke uspješno prebacili u "gh-pages". unesite sljedeću naredbu:

```
git checkout -b gh-pages
```

Za provjeru statusa datoteka unesite naredbu

```
git status
```

Vidjet ćete popis datoteka u vašem git repozitoriju. Sve će biti označene crvenom bojom i to znači da ih trebamo prebaciti (eng. push) s lokalnog na udaljeni SITblog repozitorij. Za početak krenimo s dodavanjem (eng. add).

```
git add .
```

Nakon toga prebacimo sve sadržaje u repozitorij.

```
git commit -m "inicijalni commit"
```

Pazite na **-m** – jako bitno u kasnijim fazama upravljanja promjenama sadržaja.

Pronađite na Githubu poveznicu na udaljeni repozitorij i unesite sadržaj sličan niže navedenom retku (pozor! poveznica se razlikuje od računa do računa, od osobe do osobe, naziva repozitorija i sl.)

```
git remote add origin https://github.com/fpehar/SITblog.git
```

Sad smo spremni "gurnuti" sve datoteke u "gh-pages" ogranak udaljenog repozitorija. Unesite

```
git push origin gh-pages
```

Sustav vas može tražiti unos Github korisničkih podataka. U tom slučaju je dovoljno je unijeti korisničko ime i zaporku.

I gotovo! Sad biste sve svoje sadržaje trebali vidjeti sve svoje sadržaje na *remote* repozitoriju.

Za pristup objavljenom web sjedištu potrebno je otvoriti postavke (eng. Setting) i pronaći sekciju 'Github Pages'. Odaberite ponuđenu poveznicu i otvorit će vam se vaša nova stranica.

Ekstra važna napomena:

Obvezno u **_config.yml** datoteci dodati sljedeće postavke:

- **baseurl**: **"/SITblog"** # the subpath of your site, e.g. /blog

- `url: "https://fpehar.github.io/"` # the base hostname & protocol for your site, e.g. `http://example.com`
- `github_username: fpehar`

4. korak: Dodajte nove sadržaje na blog

Sad ste spremni unositi promjene na svom blogu. Svaki put kad napravite neku promjenu na lokalnom računalu potrebno je pokrenuti `jeekyll serve` naredbu i provjeriti izmjene. Ako ste zadovoljni s lokalnim izmjenama ponovite postupak dodavanja (`git add .`), učitavanja (`git commit -m "poruka"`), te u konačnici slanja sadržadja s lokalnog na udaljeni repozitorij, točnije u ogranak "gh-pages". I to je sve za danas.

5. korak: Q&A sesija

Za sva eventualna pitanja se vidimo na MS Teams.